REMARKS

Status of Claims

At the time the Office Action was mailed, claims 1-30 were pending in the application. In the present response, claim 4 has been amended. No claims have been added or canceled. No new matter has been added. Therefore, claims 1-30 remain pending in the application. Applicant respectfully requests reconsideration of the present application.

Claim Rejections - 35 U.S.C. § 112

Claim 4 has been rejected under 35 USC 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Examiner has noted that the limitations "the register," "the processor identification instruction" and "the selected value" in claim 4 lack sufficient antecedent basis in claim 1, from which it depends. Claim 4 has been amended to depend from claim 3, as originally intended by the applicant, which provides sufficient antecedent basis for the cited limitations. Applicant respectfully submits that the rejection has been overcome and respectfully requests the Examiner to withdraw the rejection.

Claim Rejections - 35 U.S.C. § 102

Claims 1-3, 5, 9-12, 16, 18, 20-22, and 26-30 have been rejected under 35 USC 102(e) as being anticipated by Chalmers et al, Publication No. US 2003/0182545 (hereinafter "Chalmers").

Claims 1-3 and 9-11

As to independent claims 1 and 9, applicant respectfully disagrees with the Examiner because the cited reference does not disclose each and every limitation of independent claims 1 and 9. In particular, applicant submits that Chalmers does not disclose "verifying if a brand string feature is supported on a processor."

Chalmers discloses a "BIOS code 110 and a configuration database 120 for discovering and configuring the surrounding computer system" (Chalmers, fig. 1 and page 1, para. 0016, lines 1-5) and a directory 260 in the configuration database which contains a pointer to an entry for processor speed (Chalmers, fig. 2 and page 2, para. 0019, lines 1-8). That is, Chalmers discloses a method that assumes that the required information is available in the configuration database without "verifying if a brand string feature is supported on [the] processor" as recited in independent claims 1 and 9. Accordingly, applicant respectfully submits that independent claims 1 and 9 are not anticipated by Chalmers.

Given that claims 2 and 3 depend from claim 1, and that claims 10 and 11 depend from claim 9, applicant submits that claims 2, 3, 10 and 11 are also patentable over Chalmers.

Claims 5-8, 12, 16, 18, 20-22 and 26-30

As to independent claims 5, 12, 18 and 22, applicant respectfully disagrees with the Examiner because the reference does not discloses any of the limitations of the subject claims as purported by the Office Action.

First, the Office Action purports that the limitation "loading a register with a first specified value" is disclosed by Chalmers in paragraph 0024, without explaining where the limitation is found. Chalmers discloses:

> In one embodiment, a CDB initially resides in a non-volatile or persistent memory, such that it may be expected to survive a reset or reboot of a system. However, the CDB may be copied into volatile memory such as RAM and manipulated, and may be copied back into non-volatile memory thereafter. In such an instance, d-nodes and entries created while the CDB is stored in RAM may be tagged with a volatile or non-volatile tag, indicating whether the d-node or entry should be copied when the CDB is copied back into non-volatile storage. When the CDB is used for storing configuration information of a system, some of that information may be deemed too dynamic to be worth saving in non-volatile storage, whereas other information may be sufficiently static that it should be saved.
>
> (Chalmers, page 2, para. 0024).

The reference discloses a configuration database maintained in random access memory, which is not the same as "loading a register with a first specified value." The applicant to see where the Examiner finds such a limitation in the cited reference and respectfully requests that the Examiner either point out with greater specificity where in the reference the Examiner believes the subject limitation is disclosed, or withdraw the rejections.

Second, the Office Action purports that the limitation "executing a processing instruction" is disclosed by Chalmers in paragraph 0021 without explaining where the limitation is found in the cited text. Chalmers discloses:

> In one embodiment, navigation within the CDB and setting and retrieving values from the CDB may be done with a set of functions similar to those familiar for navigating through directory structures in file systems. To begin with, a pointer to something within the CDB must be available. In one embodiment, a pointer to the root directory d-node of the CDB is always maintained as well. To move to a new directory, a cdbsetCWD (CWD--current working directory) function may be used. When supplied with the absolute path to the desired location or a relative path from the current location to the desired location, along with the current

> location, the cdbsetCWD function returns a pointer to the desired location if the supplied pathname was correct. In one embodiment, the parent d-node is defined as the d-node which has its subordinate d-node pointer pointing to the list of d-nodes in which the current d-node is found, and that parent d-node may be found with the relative path `..` from the current d-node. This allows for navigation from a d-node up the hierarchical structure. Furthermore, each directory identifier is separated by a `/` in one embodiment, thus allowing for navigation through multiple directory d-nodes to reach a desired destination. Thus, if one started with a CWD of `/proc/P0` and wanted to move to `/mem` two options would exist for specifying the desired path. First, the absolute path `/mem` could be supplied. Second, the relative path `./../mem` could be supplied.
>
> (Chalmers, page 2, para. 0021).

Here, Chalmers discloses a methodology for navigating a directory structure, which is not the same as "executing a processing instruction." Again, applicant respectfully requests that the Examiner either point out with greater specificity where in the reference the Examiner believes the subject limitation is disclosed, or withdraw the rejections.

Third, the Office Action purports that the limitations "verifying that the returned value in the register logically anded with a second specified value does not equal zero," and "verifying that a value the processing instruction returns is greater than or equal to a third specified value" are disclosed in paragraphs 0027 through 0029 of Chalmers without explaining where the limitations are found in the cited text. Chalmers discloses:

> At block 340, the results of all three independent operations are compared, thus making a determination as to whether the current system configuration, the expected system configuration and the startup system configuration match. The results of this comparison are reported at block 350, and a decision may be made either by a software algorithm or an operator of the system as to whether the CDB needs to be updated at block 360. If it is determined that the CDB needs to be updated, then at block 370 changes are made in the CDB stored in randomly accessible storage, and the CDB is then copied back to the non-volatile storage where it originally resided and where it is maintained by the BIOS. After processing at block 370, or at block 360 if no changes were necessary, the process stops at termination block 380.

> It will be appreciated that this process of comparing the CDB maintained by the system with either an expected configuration of the system or a dynamically determined configuration of the system may yield much useful information about the system. It may become apparent that the BIOS is not properly initializing the system or discovering all of the components of the system at startup. It may also become apparent that other software besides the BIOS is altering the configuration of the system, either in the CDB, or without updating the CDB. Furthermore, it may become apparent that portions of the system are malfunctioning or not performing as predicted. All of this information may prove useful in debugging a system, designing a new system, or verifying the proper performance of a system.
>
> Also, it will be appreciated that the process of constructing a CDB or CDB form for information may be executed using the data structures and functions described previously with respect to FIG. 2, as the data structure may be created, traversed, and compared using the functions mentioned. As a result, an automated comparison of the expected, actual, and startup values for the configuration of the system may be made in a straightforward manner, and the differences between two or three CDB structures may be reported in a manner useful for human or automated interpretation.
>
> (Chalmers, page 3, paras. 0027-0029).

In the cited text, Chalmers discloses the structure and function of a configuration database, which has nothing to do with "verifying that the returned value in the register logically anded with a second specified value does not equal zero," and "verifying that a value the processing instruction returns is greater than or equal to a third specified value."

Again, the applicant respectfully requests that the Examiner either point out with greater specificity where in the reference the Examiner believes the subject limitations are disclosed, or withdraw the rejections.

Fourth, the Office Action purports that the limitation "scanning the string in reverse order for at least one specified substring" is disclosed in paragraphs 0014 and 0029 of Chalmers, without explaining where the limitations are found in the cited text. Chalmers discloses:

> A method and apparatus for a register scan process is described. In the following description, for purposes of explanation, numerous specific details are set forth

> in order to provide a thorough understanding of the invention. It will be apparent, however, to one skilled in the art that the invention can be practiced without these specific details. In other instances, structures and devices are shown in block diagram form in order to avoid obscuring the invention.
>
> (Chalmers, page 1, para. 0014).
>
> Also, it will be appreciated that the process of constructing a CDB or CDB form for information may be executed using the data structures and functions described previously with respect to FIG. 2, as the data structure may be created, traversed, and compared using the functions mentioned. As a result, an automated comparison of the expected, actual, and startup values for the configuration of the system may be made in a straightforward manner, and the differences between two or three CDB structures may be reported in a manner useful for human or automated interpretation.
>
> (Chalmers, page 3, para. 0029).

The cited paragraphs, far from disclosing the limitation "scanning the string in reverse order for at least one specified substring," amount to disclaimers of any specific methods of scanning registers. Again, the applicant respectfully requests that the Examiner either point out with greater specificity where in the reference the Examiner believes the subject limitation is disclosed, or withdraw the rejection.

Finally, the Office Action purports that the limitations "parsing the next digits as a decimal number," "determining a multiplier value according to the specified substring," and "multiplying the decimal number by the multiplier value to output a maximum operating frequency" are disclosed in paragraphs 0019 and 0025-0029 of Chalmers, without explaining where the limitations are found in the cited text. Chalmers discloses:

> P0 directory 260 contains the identifier `P0`, a pointer to a peer d-node for P1 directory 270, and a pointer to an entry for processor ID entry 265. Processor ID entry 265 contains a type indicating what type of entry is stored (a string in this instance), a value which holds the value of the entry (`PIII001` in this instance), an ID indicating the identification of the entry (`Processor ID` in this instance) and a pointer to another entry for processor speed entry 268. Processor speed entry 268 contains a type indicating what type of entry is stored (a string in this

instance), a value which holds the value of the entry (`Auto` in this instance), an ID indicating the identification of the entry (`Processor Speed` in this instance), and a pointer to another entry which is not shown.

(Chalmers, page 2, para. 0019).

Turning to FIG. 3, a flow diagram of a process of evaluating a system's configuration using a CDB is illustrated. Initially, the process starts at start block 300. Three independent sets of operations branch out from start block 300, and these three sets of operations may occur in a serial or parallel fashion. Furthermore, only two of these sets of operations need occur, as the object is to compare what the startup of the computer yields as a system configuration with an expected or actual configuration. At block 310, the current system configuration is determined, either by polling the parts of the system or by otherwise examining the system, typically through use of software similar to that used by the BIOS to evaluate the system at startup. If necessary, block 315 involves transforming the current system configuration into a CDB form, by creating a d-node and entry structure along the lines of that outlined with respect to FIG. 2. At block 320, the CDB as developed and maintained by the BIOS is copied to RAM or some form of randomly accessible storage. At block 330, the expected values of the CDB are obtained, and this may occur by having a person who has evaluated the system enter these values in some manner, or by using a copy of an earlier generated CDB. At block 335, the expected CDB values are transformed into a CDB form if necessary.

In one embodiment, a CM (configuration manager) is used to copy the CDB to RAM, to check the current configuration of the system, and to receive the expected configuration from the operator. The CM is a software tool using the functions described with respect to FIG. 2 to maintain and manipulate the CDB, and is designed with a user interface allowing a user to determine what the current configuration of the system is. It will be appreciated that the CM may be implemented in a manner that is highly dependent on the underlying system, and that while some aspects of the CM may be generic, most aspects of the CM will thus be system-dependent.

At block 340, the results of all three independent operations are compared, thus making a determination as to whether the current system configuration, the expected system configuration and the startup system configuration match. The results of this comparison are reported at block 350, and a decision may be made either by a software algorithm or an operator of the system as to whether the CDB needs to be updated at block 360. If it is determined that the CDB needs to be updated, then at block 370 changes are made in the CDB stored in randomly accessible storage, and the CDB is then copied back to the non-volatile storage where it originally resided and where it is maintained by the BIOS. After

processing at block 370, or at block 360 if no changes were necessary, the process stops at termination block 380.

It will be appreciated that this process of comparing the CDB maintained by the system with either an expected configuration of the system or a dynamically determined configuration of the system may yield much useful information about the system. It may become apparent that the BIOS is not properly initializing the system or discovering all of the components of the system at startup. It may also become apparent that other software besides the BIOS is altering the configuration of the system, either in the CDB, or without updating the CDB. Furthermore, it may become apparent that portions of the system are malfunctioning or not performing as predicted. All of this information may prove useful in debugging a system, designing a new system, or verifying the proper performance of a system.

Also, it will be appreciated that the process of constructing a CDB or CDB form for information may be executed using the data structures and functions described previously with respect to FIG. 2, as the data structure may be created, traversed, and compared using the functions mentioned. As a result, an automated comparison of the expected, actual, and startup values for the configuration of the system may be made in a straightforward manner, and the differences between two or three CDB structures may be reported in a manner useful for human or automated interpretation.

(Chalmers, page 2, para. 0025 through page 3, para. 0029).

The Office Action purports that the subject limitations can be found in this 856-word passage from Chalmers, cited by the Examiner. After a review of the passage, the applicant at a loss to discern where in the cited reference the subject limitations may be disclosed. Again, the applicant respectfully requests that the Examiner either point out with greater specificity where in the reference the Examiner believes the subject limitations are disclosed, or withdraw the rejections.

Claim Rejections - 35 U.S.C. § 103

Claims 4, 6-8,13-15,17, 19, 23-25 have been rejected under 35 USC 103(a) as being unpatentable over Chalmers.

The applicant respectfully refers the Examiner to the foregoing arguments regarding independent claims 1, 5, 12, 18 and 22. Given that claim 4 depends indirectly from claim 1, that claims 6-8 depend from claim 5, that claims 13-15 depend from claim 12, that claim 19 depends from claim 18 and that claims 23-25 depend from claim 22, applicant submits that the subject claims are patentable over Chalmers.

Conclusion

In view of the foregoing amendments and arguments, the applicant submits that all objections and rejections have been traversed and that the application is in condition for allowance. Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

If there are any additional charges, please charge Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 8/26, 2004

Richard W. Thill
Reg. No. 53,686

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300